HCTR: A Variable-Input-Length Enciphering Mode

Peng Wang[1], Dengguo Feng[1,2], and Wenling Wu[2]

[1] State Key Laboratory of Information Security,
Graduate School of Chinese Academy of Sciences, Beijing 100049, China
w.rocking@gmail.com

[2] State Key Laboratory of Information Security,
Institution of Software of Chinese Academy of Sciences, Beijing 100080, China
{feng, wwl}@is.iscas.ac.cn

Abstract. This paper proposes a blockcipher mode of operation, HCTR, which is a length-preserving encryption mode. HCTR turns an n-bit blockcipher into a tweakable blockcipher that supports arbitrary variable input length which is no less than n bits. The tweak length of HCTR is fixed and can be zero. We prove that HCTR is a strong tweakable pseudorandom permutation ($\widetilde{sprp}$), when the underlying blockcipher is a strong pseudorandom permutation ($sprp$). HCTR is shown to be a very efficient mode of operation when some pre-computations are taken into consideration. Arbitrary variable input length brings much flexibility in various application environments. HCTR can be used in disk sector encryption, and other length-preserving encryptions, especially for the message that is not multiple of n bits.

Keywords: Blockcipher, tweakable blockcipher, disk sector encryption, modes of operation, symmetric encryption.

1 Introduction

Basic encryption modes, such as CBC [27], increase the message length. But in many scenarios, we need a length-preserving encryption (enciphering). For example, in networking application, some packet format was not defined for cryptographic purposes, and can not be altered. So when we want add privacy features, we can not even lengthen one bit. The other example is disk sector encryption. A disk is partitioned into fixed-length sectors. The sector-level encryption is a low-level encryption. The encryption device knows nothing about the information of files or directories. It encrypts or decrypts sectors when they arrive. Suppose the plaintext at the sector location of T is P, and the encryption algorithm is $\widetilde{E}$, then the ciphertext stored in this sector is $C = \widetilde{E}_K^T(M)$, where K is the secret key. Of course we can not expand the message length, so $|M| = |\widetilde{E}_K^T(M)|$. That is why we need the concept of *tweakable blockcipher* in disk sector encryption. The sector location T is call *tweak*, which is also called associated data in [15, 13].

D. Feng, D. Lin, and M. Yung (Eds.): CISC 2005, LNCS 3822, pp. 175–188, 2005.

In the above example the message length is not always fixed and the same as, but usually much longer than, that of well known blockciphers such as DES (64 bits) or AES (128 bits) [6]. For example the sector length is typically 512 bytes. So we need wide-block-length enciphering modes based on blockciphers. When we have a wide-block-length enciphering mode, we can easily put the tweak into it using the method in [11] or [8], to get a tweakable enciphering mode.

This paper proposes a tweakable enciphering mode, or an arbitrary-variable-input-length tweakable blockcipher. We name it HCTR, for it makes use of a special universal hash function and the CTR mode. If the underlying blockcipher is $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, then our mode supports arbitrary variable length of at lest n bits, using a $(k+n)$-bit key and m blockcipher calls to encipher m blocks plaintext. The length of tweak in HCTR is fixed and can be zero. When it is zero, HCTR becomes an enciphering mode, or a arbitrary-variable-input-length blockcipher.

Our HCTR mode is a hash-encipher-hash construction, part of the middle layer uses the CTR encryption mode. HCTR is similar to the XCB mode [13], and also can be viewed as a generalization to the basic construction of $\widetilde{sprp}$ in [11]. The ABL mode [15] and the XCB mode [13] are unbalanced Feistel constructions using universal hash functions as their components. They also support variable input length, but the secret key is very long (4 keys in ABL and 5 keys in XCB) and have to be generated from a main key. The CMC mode [8] and the EME mode [9] are modes without using any universal hash functions. But they only support the message that is multiple of a block. HCTR has great advantage among these modes.

The attack-model is an adaptive chosen plaintext/ciphertext attack: an adversary can choose a tweak T, a plaintext P and get a ciphertext $C = \widetilde{E}_K^T(P)$; or choose a tweak T, a ciphertext C and get a plaintext $P = (\widetilde{E}_K^T)^{-1}(C)$. The current query can base on previous answers. We prove that HCTR is a strong secure tweakable blockcipher ($\widetilde{sprp}$), which is defined as the one indistinguishable from the independently random permutations indexed by the tweak T. If HCTR is used in disk sector encryption, the effect is that each sector is encrypted with a different random permutation independently. This kind of tweakable blockcipher is under standardization [19] by the IEEE Security in Storage Working Group. Our proof method adopts the game-play technique [2, 26], which was first used in [10].

We give basic definitions in Section 2. Specification of HCTR is in Section 3. Section 4 discusses some insure modifications and compares HCTR with other modes. The concrete security bound is given in Section 5.

1.1 Related Work

Constructions of large-block-size blockciphers from small-block-size blockciphers can date back to the pioneering work of Luby and Rackoff [12]. They showed that three rounds of the Feistel structure turns n-bit to n-bit random functions into a $2n$-bit secure blockcipher, and four rounds into a strong secure one. Naor and Reingold [18] showed that two rounds Feistel construction with initial and final strong universal invertible hash functions is enough to construct a strong secure blockcipher. In [17], they further used this hash-encipher-hash construction to get a mode of operation, but the hash function is quiet complex. Patel etc. further discussed the function of universal hash functions in the Feistel construction [20]. Bellare and Rogaway [1] used a special pseudorandom function and a special encryption mode to construct a variable-input-length cipher. Patel etc. [21] made some efficiency improvement to this scheme and the other unbalanced Feistel construction by using universal hash functions.

The constructions of tweakable blockciphers from scratch involve HPC [24] and Mercy [5] (although it has been broken by Fluhrer [7]).

Tweakable blockcipher is not only a suitable model for disk sector encryption and useful in length-preserving encryption, but also a good starting point to do design problem [11]. Following this thought, Rogaway [22] made refinement to modes OCB [23] and PMAC [3] using tweakable blockciphers.

2 Basic Definitions

Blockciphers and tweakable blockciphers. A *blockcipher* is a function $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where $E_K(\cdot) = E(K, \cdot)$ is a *length-preserving* permutation for all $K \in \mathcal{K}$. $\mathcal{K} \neq \phi$ is a *key space* and $\mathcal{M} \neq \phi$ is a *message space.* A *tweakable blockcipher* is a function $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\widetilde{E}_K^T(\cdot) = \widetilde{E}_K(T, \cdot) = \widetilde{E}(K, T, \cdot)$ is a *length-preserving* permutation for all $K \in \mathcal{K}$ and $T \in \mathcal{T}$. $\mathcal{T}$ is a *tweak space.*

We write $s \xleftarrow{R} S$ to denote choosing a random element s from a set S by uniform distribution. Let $\mathrm{Perm}(\mathcal{M})$ be the set of all length-preserving permutations on $\mathcal{M}$. When $\mathcal{M} = \{0,1\}^n$, we denote it as $\mathrm{Perm}(n)$. Let $\mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ be the set of all mappings from $\mathcal{T}$ to $\mathrm{Perm}(\mathcal{M})$. $\mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$ can also be viewed as the set of all blockciphers $E : \mathcal{T} \times \mathcal{M} \to \mathcal{M}$. If $\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$, then for every $T \in \mathcal{T}$, $\widetilde{\pi}^T(\cdot) = \widetilde{\pi}(T, \cdot)$ is a random permutation. When $\mathcal{M} = \{0,1\}^n$, we denote it as $\mathrm{Perm}^{\mathcal{T}}(n)$.

An *adversary* is a (randomized) algorithm with access to one or more oracles which are written as superscripts. Without loss of generality, we assume that adversaries never ask trivial queries whose answers are already known. For example, an adversary never repeats a query and never asks $(\widetilde{E}_K)^{-1}(T, C)$ after receiving C as an answer to $\widetilde{E}_K(T, M)$, and so forth. Let $A^\rho \Rightarrow 1$ be the event that adversary A with oracle ρ outputs the bit 1.

$\widetilde{prp}$ and $\widetilde{sprp}$. A tweakable blockcipher $\widetilde{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ is a *(strong) pseudorandom tweakable permutation* ($\widetilde{prp}$ or $\widetilde{sprp}$), if it is indistinguishable from a random tweakable permutation $\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$. More specifically, if the advantage function

$$\begin{aligned}\mathbf{Adv}_{\widetilde{E}}^{\widetilde{prp}}(A) = \Pr[K \xleftarrow{R} \mathcal{K} : A^{\widetilde{E}_K(\cdot,\cdot)} \Rightarrow 1] \\ - \Pr[\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}) : A^{\widetilde{\pi}(\cdot,\cdot)} \Rightarrow 1]\end{aligned}$$

is sufficiently small for any A with reasonable resources, then $\widetilde{E}$ is said to be a *pseudorandom tweakable permutation* ($\widetilde{prp}$), or a secure tweakable blockcipher, or secure against chosen plaintext attack. If the advantage function

$$\begin{aligned}\mathbf{Adv}_{\widetilde{E}}^{\widetilde{sprp}}(A) &= \Pr[K \xleftarrow{R} \mathcal{K} : A^{\widetilde{E}_K(\cdot,\cdot),\widetilde{E}_K^{-1}(\cdot,\cdot)} \Rightarrow 1] \\ &\quad - \Pr[\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}) : A^{\widetilde{\pi}(\cdot,\cdot),\widetilde{\pi}^{-1}(\cdot,\cdot)} \Rightarrow 1]\end{aligned}$$

is sufficiently small for any A with reasonable resources, then $\widetilde{E}$ is said to be a *strong pseudorandom tweakable permutation* ($\widetilde{sprp}$), or a strong secure tweakable blockcipher, or secure against chosen ciphertext attack.

prp AND *sprp*. When the tweak space $\mathcal{T} = \phi$, the tweakable blockcipher becomes the blockcipher. A blockcipher $E : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is a *(strong) pseudorandom permutation* (*prp* or *sprp*), if it is indistinguishable from a random permutation $\pi \xleftarrow{R} \mathrm{Perm}(\mathcal{M})$. $\widetilde{prp}$ and $\widetilde{sprp}$ correspond to *prp* and *sprp* respectively.

3 Specification of HCTR

3.1 Notations

A *string* is a finite sequence of symbols, each symbol being 0 or 1. A *block* is a string of fixed length. The blockcipher and multiplication of the finite field are operations over blocks. Let $\{0,1\}^*$ be the set of all strings. If $X, Y \in \{0,1\}^*$, then $X||Y$ is their concatenation. If $X \in \{0,1\}^*$, then the *bit-length* of X, denoted as $|X|$, is the number of bits in X. $|X| = 0$ if and only if X is the empty string ε. If one block is n bits, we can parse X into $m = \lceil |X|/n \rceil$ blocks: $X = X_1, \cdots, X_m$, where $|X_m| \leq n$, and $|X_1| = \cdots = |X_{m-1}| = n$. Let $|X|_n = \lceil |X|/n \rceil$. We say that $|X|$ has $|X|_n$ blocks. $X[s]$ denotes the s^{th} bit of X from left to right. $X[s,t]$ denotes the substring from the s^{th} bit to the t^{th} bit in X from left to right. For example, if $X = 110011$, then $X[2,4] = 100$. If $X, Y \in \{0,1\}^*$, then $X \leftarrow\!\!\!\!\oplus\, Y$ is slightly different to $X \oplus Y$. If $|X| < |Y|$ then $X \leftarrow\!\!\!\!\oplus\, Y = X \oplus Y[1, |X|]$. If $|X| = |Y|$ then $X \leftarrow\!\!\!\!\oplus\, Y = X \oplus Y$. If $|X| > |Y|$ then $X \leftarrow\!\!\!\!\oplus\, Y = X \oplus Y0^*$.

3.2 Multiplication in $GF(2^n)$

We interchangeably think of a block $L = (L_1, \cdots, L_n)$ as an abstract point in the finite field $GF(2^n)$ and as a polynomial $L(x) = L_1 + L_2x + \cdots + L_nx^{n-1}$ in $GF(2)[x]/(p(x))$, where $p(x)$ is an irreducible polynomial of degree n in $GF(2)[x]$. The addition in $GF(2^n)$ is bitwise xor $\oplus$. The multiplication of $A, B \in GF(2^n)$ is denoted as $A \cdot B$ which can be calculated as $A(x)B(x)$ in $GF(2)[x]/(p(x))$. If we choose the blockcipher as AES [6], then the bit-length of a block is 128 bits. The corresponding irreducible polynomial can be chosen as $p(x) = 1 + x + x^2 + x^7 + x^{128}$.

3.3 Universal Hash Function

H is a function family: $H = \{H_h : \{0,1\}^* \rightarrow \{0,1\}^n | h \in \{0,1\}^n\}$. For any $X \in \{0,1\}^*$, X is padded into complete blocks and then the polynomial evaluation [4] is used. Suppose $|X|_n = m$, we parse X into $X = X_1, \cdots, X_m$. We append 0s, possibly none, at the end of X to complete the block and append $|X|$, which is written as a n-bit string, as the last block. Then we use polynomial evaluation hash function in h on the padding result. More specifically, H_h is defined as:

$$H_h(X) = X_1 \cdot h^{m+1} \oplus \cdots \oplus X_m 0^* \cdot h^2 \oplus |X| \cdot h$$

which can be calculated as following:

```
Algorithm $H_h(X)$
    parse $X$ as $X_1, \cdots, X_m$
    $Y_0 \leftarrow 0^n$
    for $i \leftarrow 1$ to $m$ do
        $Y_i \leftarrow (Y_{i-1} \oplus X_i) \cdot h$
    $Y_{m+1} \leftarrow (Y_m \oplus |X|) \cdot h$
    return $Y_{m+1}$
```

When X is empty string, we define that $H_h(X) = h$. H is a special AXU (Almost Xor Universal) hash function. It has following properties which will be used in the security proof of HCTR.

1. For any $X_1, X_2 \in \{0,1\}^*, Y \in \{0,1\}^n$ and $X_1 \neq X_2$, $H_h(X_1) \oplus H_h(X_2)$ is a nonzero polynomial in h without constant term. So $\Pr[h \xleftarrow{R} \{0,1\}^n : H_h(X_1) \oplus H_h(X_2) = Y] \leq l/2^n$, where $l = max\{|X|_n, |Y|_n\} + 1$. In other words, H is a $l/2^n$-AXU hash function.
2. For any $X, Y, Z \in \{0,1\}^*$, $|X| = |Y|$, we have $H(X) \oplus H(Y) \oplus H(Z)$ is a nonzero polynomial in h without constant term.

3.4 The CTR Mode

In HCTR we use a special form of the CTR mode:

```
Algorithm $\mathrm{CTR}_K^S(N)$
    $Y \leftarrow E_K(S \oplus 1) || \cdots || E_K(S \oplus m - 1)$
    $D \leftarrow N \oplus Y$
    return $D$
```

where $|N|_n = m - 1$, K is the key and S is the counter.

3.5 The HCTR Mode

The HCTR mode makes use of a blockcipher E and the special universal hash function H. Assume that the blockcipher is $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$. Then HCTR$[E, H]$ is

$$\mathrm{HCTR}[E, H] : \{0,1\}^{k+n} \times \{0,1\}^t \times \{0,1\}^{\geq n} \rightarrow \{0,1\}^{\geq n}$$

where $\{0,1\}^{\geq n} = \cup_{m \geq n} \{0,1\}^m$ and $t \geq 0$.

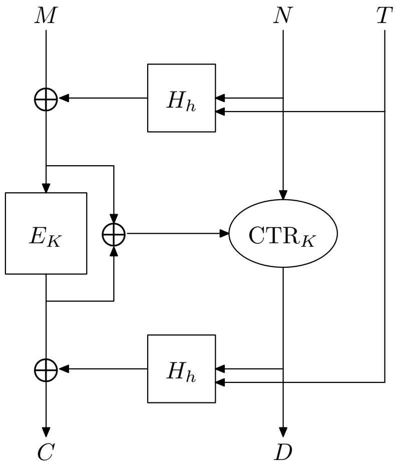

Fig. 1. The HCTR Mode

HCTR$[E, H]$ is illustrated in figure 1. we split the plaintext/ciphertext into two strings. One is the left n bits, and the other is the rest. We assume that plaintext/ciphertext has m blocks. More specifically, HCTR is the following algorithm.

```
Algorithm $\mathrm{HCTR}^T_{K,h}(M, N)$
    $MM \leftarrow M \oplus H_h(N||T)$
    $CC \leftarrow E_K(MM)$
    $S \leftarrow MM \oplus CC$
    $D \leftarrow \mathrm{CTR}^S_K(N)$
    $C \leftarrow CC \oplus H_h(D||T)$
    return $(C, D)$

Algorithm $(\mathrm{HCTR}^T_{K,h})^{-1}(C, D)$
    $CC \leftarrow C \oplus H_h(D||T)$
    $MM \leftarrow E^{-1}_K(CC)$
    $S \leftarrow MM \oplus CC$
    $N \leftarrow \mathrm{CTR}^S_K(D)$
    $M \leftarrow MM \oplus H_h(N||T)$
    return $(M, N)$
```

4 Discussions

Universal hash function. H in HCTR is a special AXU hash function. We can not substitute H by a general AXU hash function. We define a different universal hash function $H'_h(X)$ base on which HCTR is not secure. The main difference is the padding rule. In the HCTR mode, the padding rule is to append 0s and then the bit-length of X as in H. Now we first append 1 and then 0s to turn the bit-length of X into multiple of n and then use polynomial evaluation hash function. Suppose $X = X_1, \cdots, X_m$ where $|X|_n = m$, and $|X_1| = \cdots = |X_{m-1}| = n$. If $|X_m| = n$, then $H'_h(X) = X_1 \cdot h^{m+1} \oplus \cdots \oplus X_m \cdot h^2 \oplus 10^{n-1} \cdot h$. If $|X_m| < n$, then $H'_h(X) = X_1 \cdot h^m \oplus \cdots \oplus X_m 10^* \cdot h$. We can prove that $\Pr[h \xleftarrow{R} H' : h(X) \oplus h(Y) = Z] \leq \varepsilon$ for all $X, Y \in \{0,1\}^*, Z \in \{0,1\}^n, X \neq Y$. Here $\varepsilon = l/2^n$ where $l = max\{|X|_n, |Y|_n\} + 1$.

We now chose the length of tweak as 0: $\mathcal{T} = \phi$. In this situation, we can show that HCTR$[E, H']$ is not even a *prp*. We first make an arbitrary enciphering query (M^1, N^1) such that $|N^1| = n - 1$ and get an answer (C^1, D^1). If $D^1[n - 1] = N^1[n - 1]$, then we do it again until $D^1[n - 1] \neq N^1[n - 1]$. Now we make the other enciphering query (M^2, N^2) such that $M^2 = M^1 \oplus C^1$ and $N^2 = (N^1 \oplus D^1)[1, n - 2]$. We get the answer (C^2, D^2). Then the input to the second blockcipher in the last but one query is the same as the input to the first blockcipher in the last query. Therefore we have that $(N^1 \oplus D^1)[1, n-2] = (C^2 \oplus H'_h(D^2))[1, n-2]$ or $(N^1 \oplus D^1)[1, n-2] = (C^2 \oplus h \cdot D^2 10)[1, n-2]$. So we can recover h with successful probability of $1/4$ and get rid of the hash function layers. Without the hash function layers, we can easily distinguish HCTR from a random permutation.

Length of tweak. The length of tweak is fixed, because in most application environment there is no need for variable length tweak. We can chose the length of tweak according to the practical application environment. If we really need the variable length tweak, we can choose GHASH in [16, 14, 13] which is similar to H and takes two inputs.

Multiplication. The multiplication in finite field dominates the efficiency of the hash function layers. A simple implement of multiplication is even much slower than one AES call. But notice that the key h is a constant during the enciphering course, therefore we can do some pre-computations before enciphering. This time-memory tradeoffs greatly speeds up the hash function, though a bit more storage is needed. See [16, 25] for specific discussions.

	CMC	EME	ABL	XCB	HCTR
Keys	2	1	4	5	2
Blockciphers	$2m+1$	$2m+1$	$2m-2$	$m+1$	m
Universal hash	0	0	2	2	2
Variable Input Length	×	Multiple of n bits	√	√	√
Parallelizable	×	Almost	Partially	Partially	Partially

Comparisons. We compare HCTR with other enciphering modes, such as CMC, EME, ABL, and XCB, from several aspects. Suppose that we encrypt an message of m blocks. We list the comparisons in the above table. The first is the number of key. The second and third are the invocation number of the blockcipher and universal hash function. The following is whether the mode is parallelizable. In blockciphers, every bit of input bit must effect every bit of output. So there is no full parallelization. Even in the EME mode, the last layer must begin after the first layer is completely finished. In the HCTR mode, the CTR encryption can be parallelizable.

5 Security of HCTR

We prove that HCTR is a $\widetilde{sprp}$. A concrete security bound for HCTR is given in theorem 1. Lemma 1 shows that the random tweakable permutation and its inverse are indistinguishable from oracles that return random bits. This lemma greatly facilitates the proof procedure of lemma 2 which shows the security of HCTR when E_K is replaced by a random permutation.

Lemma 1 (lemma 6 in [8]). *Let* $\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M})$*. Then for any adversary A that makes q queries,*

$$\Pr[A^{\widetilde{\pi}(\cdot,\cdot),\widetilde{\pi}^{-1}(\cdot,\cdot)} \Rightarrow 1] - \Pr[A^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1] \leq q^2/2^{N+1}$$

where $\$(T, M)$ returns $|M|$ random bits and N is the bit-length of a shortest string in $\mathcal{M}$.

Let HCTR[Perm(n), H] be a variant of HCTR that uses a random permutation on n bits instead of E_K. Specifically, the key generation algorithm returns a random permutation $\pi \xleftarrow{R} \mathrm{Perm}(n)$ and a random string $h \xleftarrow{R} \{0,1\}^n$. We first give a concrete security bound for HCTR[Perm(n), H].

Lemma 2. *Let* $\mathbf{E} = \mathrm{HCTR}[\mathrm{Perm}(n), H]$*. Then for any adversary A that asks enciphering/deciphering queries totalling σ blocks,*

$$\Pr[A^{\mathbf{E}(\cdot,\cdot),\mathbf{E}^{-1}(\cdot,\cdot)} \Rightarrow 1] - \Pr[A^{\$(\cdot,\cdot),\$(\cdot,\cdot)} \Rightarrow 1] \leq ((2+t_0)\sigma^2 + \sigma^3)/2^n$$

where $\$(T, M)$ returns $|M|$ random bits and $t_0 = |T|_n$.

A proof is given in Appendix B.

We now present our result for HCTR[E, H]. Our theorem shows that if E is *sprp*, then HCTR[E, H] is a $\widetilde{sprp}$. More specifically, our theorem states that if there is an adversary A attacking the strong pseudorandomness of HCTR[E, H] asking at most σ blocks queries, then there is an adversary B attacking the strong pseudorandomness of E, such that $\mathbf{Adv}_E^{sprp}(B) \geq \mathbf{Adv}_{\mathrm{HCTR}[E,H]}^{\widetilde{sprp}}(A) - q^2/2^{n+1} - ((2+t_0)\sigma^2 + \sigma^3)/2^n$. So when $\mathbf{Adv}_E^{sprp}(B)$ is small for any B with reasonable resources, $\mathbf{Adv}_{\mathrm{HCTR}[E,H]}^{\widetilde{sprp}}(A)$ must be small. This means that the strong security of E implies the strong security of HCTR[E, H]. The theorem for HCTR[E, H] is given bellow.

Theorem 1. *For any adversary A that makes q queries totalling σ plaintext/-ciphertext blocks, there is an adversary B that makes σ queries, such that*

$$\mathbf{Adv}_{\mathrm{HCTR}[E,H]}^{\widetilde{sprp}}(A) \leq \mathbf{Adv}_E^{sprp}(B) + q^2/2^{n+1} + ((2+t_0)\sigma^2 + \sigma^3)/2^n$$

where $t_0 = |T|_n$. Furthermore, B runs in approximately the same time as A.

Proof (of theorem 1). Let $\mathbf{E}_1 = \mathrm{HCTR}[E, H]$ and $\mathbf{E}_2 = \mathrm{HCTR}[\mathrm{Perm}(n), H]$. $\widetilde{\pi} \xleftarrow{R} \mathrm{Perm}^{\mathcal{T}}(n)$ where $\mathcal{T} = \{0,1\}^t$. Consider following probabilities:

$$\begin{aligned} p_1 &= \Pr[A^{\mathbf{E}_1,\mathbf{E}_1^{-1}} \Rightarrow 1] - \Pr[A^{\mathbf{E}_2,\mathbf{E}_2^{-1}} \Rightarrow 1], \\ p_2 &= \Pr[A^{\mathbf{E}_2,\mathbf{E}_2^{-1}} \Rightarrow 1] - \Pr[A^{\$,\$} \Rightarrow 1], \\ p_3 &= \Pr[A^{\$,\$} \Rightarrow 1] - \Pr[A^{\widetilde{\pi},\widetilde{\pi}^{-1}} \Rightarrow 1]. \end{aligned}$$

Adversary B simulates A and returns whatever A returns. Then $p_1 = \mathbf{Adv}_E^{sprp}(B)$. By lemma 1, we have $p_3 \le q^2/2^{n+1}$. By lemma 2, we have $p_2 \le ((2+t_0)\sigma^2 + \sigma^3)/2^n$. □

Acknowledgment

We thank the anonymous referees for their many helpful comments. This research is supported by the National Natural Science Foundation Of China (No. 60273027, 60373047, 60025205); the National Grand Fundamental Research 973 Program of China(No. G1999035802, 2004CB318004).

References

1. M. Bellare and P. Rogaway. On the construction of variable-input-length ciphers. In L. Knudsen, editor, *Fast Software Encryption 1999*, volume 1636 of *LNCS*, pages 231–244. Springer-Verlag, 1999.
2. M. Bellare and P. Rogaway. The game-playing technique. Cryptology ePrint Archive, Report 2004/331, 2004. `http://eprint.iacr.org/`.
3. J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. In L. R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 384–397. Springer-Verlag, 2002.
4. J. L. Carter and M. N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.
5. P. Crowley. Mercy: A fast large block cipher for disk sector encryption. In B. Schneier, editor, *Fast Software Encryption 2000*, volume 1978 of *LNCS*, pages 49–63. Springer-Verlag, 2001.
6. FIPS-197. Federal information processing standards publication (FIPS 197). Advanced Encryption Standard (AES), 2001. `http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`.
7. S. R. Fluhrer. Cryptanalysis of the Mercy block cipher. In M. Matsui, editor, *Fast Software Encryption 2001*, volume 2355 of *LNCS*, pages 28–36. Springer-Verlag, 2002.
8. S. Halevi and P. Rogaway. A tweakable enciphering mode. In D. Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *LNCS*, pages 482–499. Springer-Verlag, 2003.
9. S. Halevi and P. Rogaway. A parallelizable enciphering mode. In T. Okamoto, editor, *The Cryptographers' Track at RSA Conference – CT-RSA 2004*, volume 2964 of *LNCS*. Springer-Verlag, 2004.

10. J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO 1996*, volume 1109 of *LNCS*, pages 252–267. Springer-Verlag, 1996.
11. M. Liskov, R. L. Rivest, and D. Wagner. Tweakable block ciphers. In M. Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *LNCS*, pages 31–46. Springer-Verlag, 2002.
12. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal on Computing*, 17(2):373–386, 1988. Special issue on cryptography.
13. D. A. McGrew and S. R. Fluhrer. The extended codebook (XCB) mode of operation. Cryptology ePrint Archive, Report 2004/278, 2004. `http://eprint.iacr.org/`.
14. D. A. McGrew and J. Viega. The security and performance of the galois/counter mode (GCM) of operation. In A. Canteaut and K. Viswanathan, editors, *Advances in Cryptology – INDOCRYPT 2004*, volume 3348 of *LNCS*, pages 343–355. Springer-Verlag, 2002.
15. D. A. McGrew and J. Viega. The ABL mode of operation, 2004. `http://grouper.ieee.org/groups/1619/email/pdf00004.pdf`.
16. D. A. McGrew and J. Viega. The galois/counter mode of operation (GCM), 2004. `http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes`.
17. M. Naor and O. Reingold. A pseudo-random encryption mode. `http://wisdom.weizmann.ac.il/ naor/`.
18. M. Naor and O. Reingold. On the construction of pseudo-random permutations: Luby-rackoff revisited. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (STOC '97)*, pages 189–199, New York, 1997. Association for Computing Machinery.
19. P1619. IEEE Security in Storage Working Group. `http://www.siswg.org/`.
20. S. Patel, Z. Ramzan, and G. S. Sundaram. Towards making Luby-Rackoff ciphers optimal and practical. In L. Knudsen, editor, *Fast software encryption 1999*, volume 1636 of *LNCS*, pages 171–185. Springer-Verlag, 1999.
21. S. Patel, Z. Ramzan, and G. S. Sundaram. Efficient constructions of variable-input-length block ciphers. In H. Handschuh and M. A. Hasan, editors, *Selected Areas in Cryptography 2004*, volume 3357 of *LNCS*, pages 326–340. Springer-Verlag, 2005.
22. P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. In P. J. Lee, editor, *Advances in Cryptology – ASIACRYPT 2004*, volume 3329 of *LNCS*, pages 16–31. Springer-Verlag, 2004.
23. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryptiona. In *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pages 196–205, 2001.
24. R. Schroeppel. The hasty pudding cipher. `http://www.cs.arizona.edu/rcs/hpc/`.
25. V. Shoup. On fast and provably secure message authentication based on universal hashing. In N. Koblitz, editor, *Advances in Cryptology – CRYPTO 1996*, volume 1109 of *LNCS*, pages 313–328. Springer-Verlag, 1996.
26. V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. `http://eprint.iacr.org/`.
27. SP-800-38A. Recommendation for block cipher modes of operation - methods and techniques. NIST Special Publication 800-38A, 2001. `http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf`.

A Intellectual Property Statement

The authors explicitly release any intellectual property rights to the HCTR mode into the public domain. Further, the authors are not aware of any patent or patent application anywhere in the world that cover this mode.

B Proof of Lemma 2

Proof (of lemma 2). Suppose A makes q queries. Assume that the r^{th} query is (T^r, U^r, V^r), where T^r is the tweak, (U^r, V^r) is the plaintext(ciphertext). Suppose that $m_r = |(U^r, V^r)|_n$. $\sigma = m_1 + \cdots + m_q$ is the total plaintext/ciphertext block number. Furthermore, we split V^r into blocks: $V^r = V_1^r, \cdots, V_{m_r-1}^r$. We describe the attacking procedure of A as the interaction with games.

Game 1 and **Game 2**. The following Game 1 illustrates how HCTR[Perm(n), H] and its inverse answer A's queries:

```
$\mathcal{D} \leftarrow \mathcal{R} \leftarrow \phi$; $bad \leftarrow$ false

If the $r^{th}$ query $(T^r, U^r, V^r)$ is an enciphering query:
  $UU^r \leftarrow U^r \oplus H_h(V^r || T^r)$
  $XX^r \xleftarrow{R} \{0,1\}^n$
  if $UU^r \in \mathcal{D}$ then $bad \leftarrow$ true $\boxed{XX^r \leftarrow \pi(UU^r)}$
  if $XX^r \in \mathcal{R}$ then $bad \leftarrow$ true $\boxed{XX^r \xleftarrow{R} \bar{\mathcal{R}}}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{UU^r\}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{XX^r\}$
  $S^r \leftarrow UU^r \oplus XX^r$
  for $i \leftarrow 1$ to $m_r - 1$ do
    $YY_i^r \xleftarrow{R} \{0,1\}^n$
    if $S^r \oplus i \in \mathcal{D}$ then $bad \leftarrow$ true $\boxed{YY_i^r \leftarrow \pi(S^r \oplus i)}$
    if $YY_i^r \in \mathcal{R}$ then $bad \leftarrow$ true $\boxed{YY_i^r \xleftarrow{R} \bar{\mathcal{R}}}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{YY_i^r\}$
  $YY^r \leftarrow YY_1^r || \cdots || YY_{m_r-1}^r$
  $D^r \leftarrow V^r \oplus YY^r$
  $C^r \leftarrow XX^r \oplus H_h(D^r || T^r)$
  return $(C^r, D^r)$

If the $r^{th}$ query $(T^r, U^r, V^r)$ is an deciphering query:
  $UU^r \leftarrow U^r \oplus H_h(V^r || T^r)$
  $XX^r \xleftarrow{R} \{0,1\}^n$
  if $UU^r \in \mathcal{R}$ then $bad \leftarrow$ true $\boxed{XX^r \leftarrow \pi^{-1}(UU^r)}$
  if $XX^r \in \mathcal{D}$ then $bad \leftarrow$ true $\boxed{XX^r \xleftarrow{R} \bar{\mathcal{D}}}$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{XX^r\}$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \{UU^r\}$
```

```
$S^r \leftarrow UU^r \oplus XX^r$
**for** $i \leftarrow 1$ **to** $m_r - 1$ **do**
    $YY_i^r \overset{R}{\leftarrow} \{0,1\}^n$
    **if** $S^r \oplus i \in \mathcal{D}$ **then** $bad \leftarrow$ **true** $\boxed{YY_i^r \leftarrow \pi(S^r \oplus i)}$
    **if** $YY_i^r \in \mathcal{R}$ **then** $bad \leftarrow$ **true** $\boxed{YY_i^r \overset{R}{\leftarrow} \bar{\mathcal{R}}}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{YY_i^r\}$
$YY^r \leftarrow YY_1^r || \cdots || YY_{m_r-1}^r$
$N^r \leftarrow V^r \leftarrow\!\!\!\!\!\oplus YY^r$
$M^r \leftarrow XX^r \oplus H_h(N^r || T^r)$
**return** $(M^r, N^r)$
```

Notice that the permutation π is not chosen before the attack, but "on the fly" as needed to answer the queries during the attacking procedure. The sets $\mathcal{D}$ and $\mathcal{R}$, which are multisets in which the element may repeat, keep track of the domain and the range of π respectively. Game 2 is obtained by omitting the boxed statements. Because $XX^r, XY^r (r = 1, \cdots, q)$ are independent random strings, the answers A get, when interacts with Game 2, are also independent random strings. So $A^{\text{Game 2}}$ is the same as $A^{\$,\$}$. In Game 1, each boxed statement is executed if and only if the flag *bad* is set to be true. Therefor we have

$$\begin{aligned}\Pr[A^{\mathbf{E},\mathbf{E}^{-1}} \Rightarrow 1] - \Pr[A^{\$,\$} \Rightarrow 1] \\ = \Pr[A^{\text{Game 1}} \Rightarrow 1] - \Pr[A^{\text{Game 2}} \Rightarrow 1] \leq \Pr[A^{\text{Game 2}} \text{ set } bad]. \quad (1)\end{aligned}$$

Game 3. We make some modifications to Game 2. The answer of each query is directly chosen as random string and the state of *bad* is set at the end of all queries. Game 3 is the following:

```
 Initialization :
$\mathcal{D} \leftarrow \mathcal{R} \leftarrow \phi$

On the r$^{th}$ query $(T^r, U^r, V^r)$:
   $(X^r, Y^r) \overset{R}{\leftarrow} \{0,1\}^{m_r \cdot n}$
   **return** $(X^r, Y^r)[1, |(U^r, V^r)|]$

Finalization :
**for** $r \leftarrow 1$ **to** $q$ **do**:

If the $r^{th}$ query $(T^r, U^r, V^r)$ is an enciphering query:
   $UU^r \leftarrow U^r \oplus H_h(V^r || T^r)$
   $XX^r \leftarrow X^r \oplus H_h(Y^r[1, |V^r|] \,||\, T^r)$
   $\mathcal{D} \leftarrow \mathcal{D} \cup \{UU^r\}$
   $\mathcal{R} \leftarrow \mathcal{R} \cup \{XX^r\}$
   $S^r \leftarrow UU^r \oplus XX^r$
   **for** $i \leftarrow 1$ **to** $m_r - 1$ **do**
      $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
      $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^r \leftarrow\!\!\!\!\!\oplus V_i^r\}$
```

```
If the $r^{th}$ query $(T^r, U^r, V^r)$ is a deciphering query:
    $UU^r \leftarrow U^r \oplus H_h(V^r||T^r)$
    $XX^r \leftarrow X^r \oplus H_h(Y^r[1, |V^r|] \,||\, T^r)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{XX^r\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{UU^r\}$
    $S^r \leftarrow UU^r \oplus XX^r$
    for $i \leftarrow 1$ to $m_r - 1$ do
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^r \mathbin{\leftarrow\!\!\!\!\!\oplus} V_i^r\}$

$bad \leftarrow$ (there is a repetition in $\mathcal{D}$) or (there is a repetition in $\mathcal{R}$)
```

We have

$$\Pr[A^{\text{Game 2}} \text{ set } bad] = \Pr[A^{\text{Game 3}} \text{ set } bad]. \qquad (2)$$

Without lost of generality, suppose that A is a deterministic algorithm. We want to prove that for any fixed $X^r, Y^r (r = 1, \cdots, q)$, the above probability is negligible. But that is not true. For example when $Y_1^1 \mathbin{\leftarrow\!\!\!\!\!\oplus} V_1^1 = Y_1^2 \mathbin{\leftarrow\!\!\!\!\!\oplus} V_1^2$, the *bad* is set to be true. We firstly make some restrictions on the choices of these random strings.

Restrictions on the choices of (X^r, Y^r):

1. X^r, Y_i^r are all distinct.
2. $X^r \neq U^r \oplus X^s \oplus U^s \oplus i \oplus j$ for all $s < r$, $1 \leq i \leq (m_r - 1)$, $1 \leq j \leq (m_s - 1)$.
3. $X^r \neq U^s$ for all $s < r$.
4. $Y_i^r \neq Y_j^s \mathbin{\leftarrow\!\!\!\!\!\oplus} V_j^s \mathbin{\leftarrow\!\!\!\!\!\oplus} V_i^r$, for all $s < r$ and for all $s = r, j < i$.

It is easy to calculate that each restriction in the above decreases the choices of (X^r, Y^r) at most $\sigma^2/2^{n+1}$. Totally, the choices of (X^r, Y^r) are decreased at most $2\sigma^2/2^n$.

Game 4. With these restrictions, we fix queries and answers. Suppose that $\{T^r, U^r, V^r, X^r, Y^r | r = 1, \cdots, q\}$ make the probability of setting *bad* maximum. Now consider the following non-interactive and non-adaptive Game 4:

```
for $r \leftarrow 1$ to $q$ do:

If the $r^{th}$ query $(T^r, U^r, V^r)$ is an enciphering query:
    $UU^r \leftarrow U^r \oplus H_h(V^r||T^r)$
    $XX^r \leftarrow X^r \oplus H_h(Y^r[1, |V^r|] \,||\, T^r)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{UU^r\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{XX^r\}$
    $S^r \leftarrow UU^r \oplus XX^r$
    for $i \leftarrow 1$ to $m_r - 1$ do
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^r \mathbin{\leftarrow\!\!\!\!\!\oplus} V_i^r\}$

If the $r^{th}$ query $(T^r, U^r, V^r)$ is an deciphering query:
    $UU^r \leftarrow U^r \oplus H_h(V^r||T^r)$
    $XX^r \leftarrow X^r \oplus H_h(Y^r[1, |V^r|] \,||\, T^r)$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{XX^r\}$
    $\mathcal{R} \leftarrow \mathcal{R} \cup \{UU^r\}$
```

```
        $S^r \leftarrow UU^r \oplus XX^r$
        for $i \leftarrow 1$ to $m_r - 1$ do
            $\mathcal{D} \leftarrow \mathcal{D} \cup \{S^r \oplus i\}$
            $\mathcal{R} \leftarrow \mathcal{R} \cup \{Y_i^r \oplus V_i^r\}$
    $bad \leftarrow$ (there is a repetition in $\mathcal{D}$) or (there is a repetition in $\mathcal{R}$)
```

From the above discussion, we have that

$$\Pr[A^{\text{Game 3}} \text{ set } bad] \leq \Pr[\text{Game 4 set } bad] + 2\sigma^2/2^n. \qquad (3)$$

Let $I = \{1, \cdots, q\}$, $I = I_1 \cup I_2$, where $I_1 = \{s \in I | (T^s, U^s, V^s)$ is an enciphering query$\}$ and $I_2 = \{t \in I | (T^t, U^t, V^t)$ is a deciphering query$\}$. Let $l = max\{m_1, \cdots, m_q\}$. We can see that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, where $\mathcal{D}_1 = \{UU^s | s \in I_1\}$, $\mathcal{D}_2 = \{XX^t | t \in I_2\}$, and $\mathcal{D}_3 = \{S^r \oplus i | r \in I, 1 \leq i \leq (m_r - 1)\}$. $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2 \cup \mathcal{R}_3$, where $\mathcal{R}_1 = \{XX^s | s \in I_1\}$, $\mathcal{R}_2 = \{UU^t | t \in I_2\}$, and $\mathcal{R}_3 = \{Y_i^r \oplus V_i^r | r \in I, 1 \leq i \leq (m_r - 1)\}$.

Any element in $\mathcal{D}$ or $\mathcal{R}$ is a polynomial in h whose degree is at most $(l+t_0)$. We want to prove that for any $X_1, X_2 \in \mathcal{D}$ or $X_1, X_2 \in \mathcal{R}$, the repetition probability $\Pr[X_1 = X_2] \leq (l+t_0)/2^n$. Because the polynomial of degree $(l+t_0)$ has at most $(l+t_0)$ roots in finite field, we only need to prove that $X_1 \oplus X_2$ is a nonzero polynomial in h.

We consider following situations:

- $X_1, X_2 \in \mathcal{D}_1$. (T^s, U^s, V^s), $s \in I_1$ are all distinct, because A never ask trivial query. By property 1 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1, X_2 \in \mathcal{D}_2$. By restriction 1, the constant term of $X_1 \oplus X_2$ is nonzero.
- $X_1, X_2 \in \mathcal{D}_3$. By restriction 2, the constant term of $X_1 \oplus X_2$ is nonzero.
- $X_1 \in \mathcal{D}_1$ and $X_2 \in \mathcal{D}_2$. Suppose $X_1 = UU^s$ and $X_2 = XX^t$. If $s < t$, then by restriction 3, the constant term of $X_1 \oplus X_2$ is nonzero. If $s > t$, then $(T^s, U^s, V^s) \neq (T^t, X^t, Y^t[1, |V^t|])$, because A never make trivial query. By property 1 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1 \in \mathcal{D}_2$ and $X_2 \in \mathcal{D}_3$. By property 2 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1 \in \mathcal{D}_1$ and $X_2 \in \mathcal{D}_3$. The same reason as the above.
- $X_1, X_2 \in \mathcal{R}_1$. By restriction 1, the constant term of $X_1 \oplus X_2$ is nonzero.
- $X_1, X_2 \in \mathcal{R}_2$. (T^t, U^t, V^t), $s \in I_2$ are all distinct, because A never make trivial query. By property 1 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1, X_2 \in \mathcal{R}_3$. By restriction 4, $X_1 \oplus X_2$ is a nonzero constant.
- $X_1 \in \mathcal{R}_1$ and $X_2 \in \mathcal{R}_2$. Suppose $X_1 = XX^s$ and $X_2 = UU^t$. If $s > t$, then by restriction 3, the constant term of $X_1 \oplus X_2$ is nonzero. If $s < t$, then $(T^t, U^t, V^t) \neq (T^s, X^s, Y^s[1, |V^s|])$, because A never make trivial query. By property 1 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1 \in \mathcal{R}_2$ and $X_2 \in \mathcal{R}_3$. By property 2 of H, $X_1 \oplus X_2$ is a nonzero polynomial.
- $X_1 \in \mathcal{R}_1$ and $X_2 \in \mathcal{R}_3$. The same reason as the above.

There are totally $\sigma(\sigma - 1)/2$ pairs of elements in $\mathcal{D}$ and $\sigma(\sigma - 1)/2$ pairs of elements in $\mathcal{R}$. So the probability of repetition in $\mathcal{D}$ or $\mathcal{R}$ is at most $(l+t_0)\sigma^2/2^n$.

$$\Pr[\text{Game 4 set } bad] \leq (l+t_0)\sigma^2/2^n \leq (t_0\sigma^2 + \sigma^3)/2^n. \qquad (4)$$

Combine (1), (2), (3) and (4), we complete the proof. □